

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Fitness aplikace pro domácí trenažéry s podporou sdílených dat

Fitness Application for Home Exercise Machines with Data Sharing

Zadání bakalářské práce

Student: **Marcel Jurišta**

Studijní program: **B2647 Informační a komunikační technologie**

Studijní obor: **2612R025 Informatika a výpočetní technika**

Téma: **Fitness aplikace pro domácí trenažéry s podporou sdílených dat
Fitness Application for Home Exercise Machines with Data Sharing**

Jazyk vypracování: **čeština**

Zásady pro vypracování:

Navrhněte a implementujte mobilní aplikaci pro rozšíření možností domácích fitness trenažérů o online sdílení dat. V dnešní době i ty nejlevnější fitness trenažéry disponují jednoduchým počítačem ukazujícím aktuální rychlost, uraženou vzdálenost, dobu cvičení a další údaje. Tyto údaje jsou vypočítávány na základě impulsů přicházejících ze senzoru sledující frekvenci otáček kola brzdy. Cílem bakalářské práce je mimo jiné vytvořit jednoduchý převodník, připojený přímo k této sondě, který bude posílat informace o impulsích do mobilního zařízení. Mobilní aplikace spolupracující s tímto převodníkem umožní online přenos dat na zvolené servery a umožní realtime sledování cvičení. V aplikaci bude možné zvolit režim propojení s jiným uživatelem a v realtime režimu zobrazovat srovnání jejich výkonů (virtuální závod).

1. Vyhledejte a srovnajte zařízení/řešení nabízející obdobné funkce.
2. Navrhněte a zrealizujte převodník pro přenos impulsů z fitness trenažéru do mobilního zařízení (prostřednictvím Bluetooth nebo WiFi, např. pomocí modulu ESP8266).
3. V aplikaci implementujte zpracování signálu z převodníku a přepočty obdobně, jak to realizuje dodávaný fitness trenažér, a porovnejte přesnost přepočtu.
4. Navrhněte a zrealizujte síťovou službu/protokol umožňující v realtime režimu sdílet aktuální stav s jiným uživatelem a hodnoty vhodně vizualizujte (virtuální závod).
5. Navrženou aplikaci řádně otestujte a výsledky testů kriticky zhodnoťte.

Seznam doporučené odborné literatury:

- [1] Joseph Annuzzi, Jr., Lauren Darcey, Shane Conder: Introduction to Android Application Development: Android Essentials, Addison-Wesley, 2014, ISBN 0321940261
- [2] Marco Schwartz: Internet of Things with ESP8266, Packt Publishing, 2016, ISBN 1786468026
- [3] Simon Monk: Programming Arduino: Getting Started with Sketches, Second Edition, McGraw-Hill Education TAB, 2016, ISBN 1259641635

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Mgr. Ing. Michal Krumník, Ph.D.**

Datum zadání: 01.09.2017

Datum odevzdání: 30.04.2018



doc. Ing. Jan Platoš, Ph.D.
vedoucí katedry



prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne. Uviedol som všetky literárne
pramene a publikácie, z ktorých som čerpal.

V Ostrave 16. apríla 2018

..........

Veľmi rád by som poďakoval vedúcemu bakalárskej práce **Mgr. Ing. Michalovi Krumníkovi, Ph.D.** za odbornú pomoc a konzultácie pri vytváraní tejto práce a **Bc. Michalovi Kubíkovi** za zapožičanie cyklotrenžéru pre testovanie aplikácie.

Abstrakt

Cieľom tejto bakalárskej práce je vytvorenie mobilnej aplikácie pre platformu Android, pomocou ktorej je možné vytvoriť virtuálny závod medzi viacerými cyklotrenažermi. Súčasťou práce je vytvorenie zariadenia pre prevod signálu z fitness trenažéru do mobilnej aplikácie. Používateľ sa pomocou technológie Bluetooth pripojí na zariadenie, ktorým je vopred vybavený fitness trenažér a vytvorí vlastný závod, alebo sa pripojí na už existujúci. Komunikácia medzi jednotlivými účastníkmi závodu je zabezpečená pomocou služby Google Play Games Services. Aplikácia podporuje hru maximálne 8 hráčov v jednom závode. V aplikácii je možné zahájiť aj voľný tréning, kedy používateľ sleduje iba informácie o svojej jazde. Je to zároveň jediný typ tréningu, v ktorom nie je potrebné mať nainštalovanú aplikáciu Play Games.

Kľúčové slová: Android, Bluetooth, Play Games, Google Play Games Services, fitness trenažér, virtuálny závod

Abstract

The purpose of this bachelor thesis is to create an Android mobile application, which enables creation of a virtual race between multiple stationary bicycles. The part of the work was to create an interface for transferring signals from fitness machine to mobile application. User can connect to the fitness device via Bluetooth and also can create his own race, or connect to existing one. A communication between participants is provided using Google Play Games Services. Mobile application allows multiplayer game for up to 8 participants per race. It is also possible to start a free training where the user only tracks his own ride statistics. It is the only training type, in which the user does not need to have Play Games installed.

Key Words: Android, Bluetooth, Play Games, Google Play Games Services, fitness machine, virtual race

Obsah

Zoznam použitých skratiek a symbolov	8
Zoznam obrázkov	9
Zoznam výpisov zdrojového kódu	10
1 Úvod	11
2 Obdobné riešenia	12
2.1 Exercise Along to Google Street View	12
2.2 USBCycle	13
2.3 Rouvy	13
2.4 Zwift	14
2.5 Zhrnutie	14
3 Cyklotrenažéry	15
3.1 Rozdelenie podľa ceny	15
3.2 Rozdelenie podľa spôsobu prevedenia	15
4 Prevod signálu z cyklotrenažéru	17
4.1 Arduino UNO	17
4.2 Bluetooth modul HC-05	18
4.3 Schéma zapojenia prevodníku	18
4.4 Cyklotrenažér LifeCycle 9500 HR	19
5 Vývoj mobilných aplikácií	23
5.1 Použité technológie	23
5.2 Cross-platform aplikácie	23
5.3 Java pre Android	24
5.4 Zaradenie aplikácie	25
6 Bluetooth	26
6.1 Využitie pri prenose signálu	26
6.2 Bluetooth v Android aplikácií	27
7 Návrh aplikácie	29
7.1 Aktivita	30
7.2 Intenty a Broadcast Reciever	31
7.3 Pomocné funkcie	32

7.4 Používateľské rozhranie	33
8 Google Play Games Services	35
8.1 Realtime Multiplayer	35
9 Testovanie	37
10 Záver	38
Literatúra	39

Zoznam použitých skratiek a symbolov

API	– Application Programming Interface
ART	– Android RunTime
BLE	– Bluetooth Low Energy
CSS	– Cascade Style Sheet
HTML	– Hyper Text Markup Language
I/O	– In/Out
IDE	– Integrated Development Environment
iOS	– iPhone OS
IPC	– Inter-Process Communication
JRE	– Java Runtime Environment
JVM	– Java Virtual Machine
MAC	– Media Access Control
OS	– Operačný Systém
RPM	– Revolution Per Minute
SELV	– Separated Extra Low Voltage
SMS	– Short Message Service
SPP	– Serial Port Protocol
SQL	– Structured Query Language
TTL	– Transistor Transistor Logic
USB	– Universal Serial Bus

Zoznam obrázkov

1	USBCycle a Zwift	14
2	Schéma zapojenia prevodníku	18
3	Pôvodný signál vychádzajúci z PINu RPM v porovnaní so signálom modifikova- ným pomocou optočlenu	20
4	Priamka lineárnej regresie	21
5	Zjednodušený diagram komponentov aplikácie	29
6	Hlavné menu voľba protivníkov a závod	34

Zoznam výpisov zdrojového kódu

1	Funkcia pulseIn pre meranie frekvencie na digitálnom pine	19
2	Nadviazanie spojenia s Bluetooth zariadením	28
3	Multiplayer Broadcast reciever	32
4	Serializácia objektu na String	33
5	Voľba iniciátora závodu	36

1 Úvod

Odvetvie fitness si v dnešnej dobe nachádza čoraz viac priaznivcov. Ľudia si kupujú lacné cyklotrenažéry, ktoré často poskytujú iba minimálne informácie o jazde. Vizualizácie na lacných treňažeroch sú nezáživné a zastaralé. Tréningy sú fádne a používatelia často upadajú do stereotypu. Aplikácia pre mobilné telefóny s prvkami hry, ktorá umožňuje používateľom medzi sebou súťažiť má tento stereotyp narušiť a pomocou virtuálnych závodov by sa mala dosiahnuť väčšia motivácia používateľov, zlepšenie zážitku z cvičenia a zvýšenie ich výkonov.

Aplikácia by mala podporovať jednoduché pripojenie k cyklotrenažeru a možnosť zahájenia závodu proti ostatným používateľom alebo režim voľnej jazdy, počas ktorej sú vizualizované iba informácie o jazde. Pre požadovaný výsledok je nutné vyriešiť niekoľko problémov. Tieto problémy sú rozdelené na jednotlivé úlohy a postupy ich riešení sú ďalej rozpracované.

V úvode je potrebné oboznámiť sa s obdobnými riešeniami, ktoré využívajú cyklotrenažéry pre zber informácií, analýzu a ich zobrazenie používateľsky prívetivým spôsobom. Tento prieskum má pomôcť správne navrhnúť funkčnosť výsledného produktu, ako aj inšpirovať sa populárnymi riešeniami pre vytvorenie pozitívneho používateľského zážitku. Existuje mnoho spoločností a nadšencov, ktorí sa snažia spríjemniť často nudnú a nezáživnú činnosť, ako je indoor bicyklovanie. Práve tieto projekty majú byť podkladom pre vznik tejto bakalárskej práce.

Rôzni výrobcovia cyklotrenažerov používajú rozdielne techniky pre funkčnosť brzdového systému a spracovanie a prenos informácií. Informácie, ktoré je potrebné z cyklotrenažeru získať však nie sú voľne prístupné. Znamená to, že na väčšine zariadení neexistuje rozhranie, ktoré by slúžilo na jednoduché získanie dát o jazde. Pre správny prevod informácií je potrebné vykonať analýzu dostupného cyklotrenažeru a návrh prevodníku analógového signálu z cyklotrenažeru do mobilnej aplikácie. Ďalšie kapitoly sa venujú návrhu a implementácii mobilnej aplikácie pre OS Android, ktorá má používateľovi navodiť príjemný zážitok z jazdy. Vo vypracovaní implementácie mobilnej aplikácie je dôraz kladený najmä použitiu technológie Bluetooth pre príjem dát z cyklotrenažeru, rozdeleniu aplikačnej logiky a možnosti súperenia medzi viacerými cyklistami. Hlavnou funkciou tejto aplikácie má byť práve možnosť súperenia a vytvorenia virtuálneho závodu, na čo je potrebné používať sieťovú službu pre prenos informácií v reálnom čase. V samostatnej kapitole je preto popísaný aj návrh a implementácia služieb Play Game Services pre komunikáciu viacerých používateľov v reálnom čase a vytvorenie virtuálneho závodu. V závere sa nachádza zhodnotenie reálne dosiahnutých výsledkov a reálne predpoklady pre budúci vývoj mobilnej aplikácie a využitie nových technológií.

2 Obdobné riešenia

Problém s nedostatočnou vizualizáciou a chýbajúcim nadšením pre cvičenie na stacionárnych bicykloch sa snažia riešiť domáce ale aj komerčné projekty. Najdôležitejšími činiteľmi pri produktoch, ktoré majú podporiť záujem o bicyklovanie na cyklotrenažéroch sú vhodná vizualizácia získaných dát a podpora interaktivity. Medzi dostupnými riešeniami sa nachádzajú počítačové hry, mobilné a internetové aplikácie. Dáta sú častokrát vizualizované pomocou grafov alebo virtuálnych závodov. Pre spríjemnenie zážitku z bicyklovania na cyklotrenažéri sa používajú aj záznamy cyklistov z reálnych oblastí, ktoré boli natočené práve pre tieto účely. Jedno z riešení sa snaží využiť už existujúcu počítačovú hru, ktorá je svojím nastavením prispôsobená pre imitáciu jazdy na bicykli 2.2. Prehľad obdobných riešení má byť motiváciou a inšpiráciou pre túto bakalársku prácu.

Pre ešte silnejší a reálnejší zážitok z hry sa dnes využíva veľmi populárna technológia virtuálnej reality. Hráč je posadený na špeciálne upravený stacionárny bicykel, ktorý dokáže simulovať natočenie a náklon bicykla. Má nasadené okuliare do ktorých mu je premietaná virtuálna krajina. Vďaka okuliarom na hráča nevplývajú žiadne okolité faktory a zážitok z jazdy je tak oveľa realistickejší. Táto možnosť je však oveľa nákladnejšia, ako ostatné spomínané projekty a preto sa javí ideálna iba ako atrakcia pre komerčné použitie, nie však ako aplikácia na každodenné používanie.

2.1 Exercise Along to Google Street View

Exercise Along to Google Street View je domáci projekt, ktorého hlavnou myšlienkou je čo najviac priblížiť cvičenie na cyklotrenažéri reálnemu bicyklovaniu a tak používateľovi poskytnúť dôvernejší zážitok. Pre vizualizáciu okolitej krajiny používa Google Street View a preto je možné vybrať si takmer akúkoľvek lokalitu. Pre zistenie rýchlosti sa používajú dve sondy a magnet, ktoré sú upevnené na cyklotrenažéri. Sonda je spínaná pohybom magnetu, ktorý je upevnený na pedáloch. Vývojová doska Arduino zbiera informácie o tom, akou frekvenciou sa magnet pohybuje. Posúvanie pozície na Google Street View je zabezpečené klávesovou skratkou, ktorej frekvencia spúšťania je vyvolávaná podľa aktuálnej rýchlosti pedálovania. Toto riešenie je veľmi obmedzujúce hlavne z hľadiska používateľského zážitku. Google Street View je navrhnuté tak, aby sa pri presunutí polohy znovu vykreslila okolitá krajina. Ako však môžeme vidieť aj vo videu na stránkách tohto projektu¹, prechod medzi týmito bodmi nie je plynulý a preto v človeku nedokáže vyvolať dostatočné priblíženie reality. Naskytuje sa nám však možnosť merania rýchlosti pomocou vlastného zariadenia, ktoré sníma frekvenciu otáčok kolesa. Meranie frekvencie otáčok pedálov sa javí ako dosť nepresná voľba, keďže kolesá sa pomocou zotrvačnosti točia aj keď používateľ práve nepedáluje. Informácia, ktorá reálne prichádza do vývojovej dosky Arduino je kadencia pedálovania, ktorá môže byť taktiež zaujímavá pre používateľa. Nedá sa z nej však určiť prejdená vzdialenosť, ktorá bude preto v tomto prípade nepresná a ideálnejšie sa javí možnosť snímania frekvencie otáčok kolesa. Z tohto riešenia taktiež môžeme vidieť, že vývojová

doska Arduino je dobrou voľbou pre spracovanie signálu a môže ideálne poslúžiť pre hardvérovú časť tejto baklárskej práce. Projekt je dostupný na internetovej adrese¹.

2.2 USBCycle

USBCycle je komplexný domáci projekt, ktorý pomocou dvoch vývojových dosiek Arduino Leonardo zbiera informácie o rýchlosti, natočení riadidiel a stlačení brzdy zo senzorov, ktoré sú umiestnené na bicykli. Všetky tieto informácie sú spracované a prenášané do počítačovej hry Euro Truck Simulator 2, ktorá simuluje jazdu nákladňakom. Vďaka kamere z pohľadu pred nákladňakom, kde nie je vidno jeho obrysy, vyzerá obraz dôveryhodne a tak sa používateľ cíti, akoby naozaj bicykloval vo virtuálnej krajine. Arduino transformuje signály zo senzorov a do hry sa prenášajú informácie ktoré zabezpečujú voľný pohyb po krajine, reguláciu rýchlosti a zmenu smeru jazdy. Vývojový set je do počítača pripojený ako USB zariadenie a emuluje ovládač. Pomocou rôznych tlačidiel na vývojovom sete sa dajú meniť nastavenia hry a spúšťať herné funkcie. Tento projekt potvrdil vhodnosť vývojovej dosky Arduino pre podobný typ projektov. Využitie senzoru pre rýchlosť kadencie pedálovania sa rovnako ako pri projekte 2.1 ukazuje ako nepresné, no jeho umiestnenie na koleso sa javí ako najjednoduchšia možnosť merania rýchlosti. Projekt je dostupný na internetovej adrese².

2.3 Rouvy

Ide o veľmi podarenú vylepšenú verziu projektu Exercise Along to Google Street View a rieši takmer všetky jeho nedostatky. Keďže ale ide o komerčný projekt, k tomu, aby ste túto aplikáciu mohli začať používať, musíte vynaložiť nemalé finančné prostriedky a úsilie. Spoločnosť, ktorá túto aplikáciu vyvíja neposkytuje hardvérové prostriedky pre jej funkčnosť a je potrebné zakúpiť si ich u jej partnerov. Ak aj vlastníte bicykel alebo cyklotrenažér, stále potrebujete zariadenie, ktoré zabezpečuje prenos informácií o jazde do aplikácie. Najpopulárnejšími zariadeniami sú tzv. tréner, ktoré sú však veľmi drahé a pohybujú sa v hodnote stoviek dolárov. Tréner je zariadenie, ktoré je fyzicky spojené s bicyklom a poskytuje čítanie a odosielanie informácií o rýchlosti alebo dokonca regulovanie záťaže, čím sa utvrdzuje zážitok z virtuálnej jazdy. Lacnejšou alternatívou je napríklad senzor Wahoo³, ktorý podľa použitia dokáže identifikovať otáčky kolesa alebo kadenciu pedálovania. Obe informácie môžu byť pre používateľa zaujímavé, avšak informácia o rýchlosti je častokrát skreslená a reálne na ňu vplýva mnoho faktorov. Pre používanie aplikácie Rouvy je navyše nutné platiť mesačné paušálne výdavky. Cene však odpovedá aj kvalita a aplikácia má k dispozícii veľké množstvo nahraných oblastí, z ktorých si používateľ môže vybrať. Používateľ sa pri pedáľovaní pozerá na reálne video cyklistu, ktoré bolo natočené vo vybranej oblasti. Môže sa tak virtuálne vybrať do rôznych oblastí sveta bez toho, aby bol vystavený nepriaznivým

¹<https://hackaday.com/2010/10/29/exercise-along-to-google-street-view/>

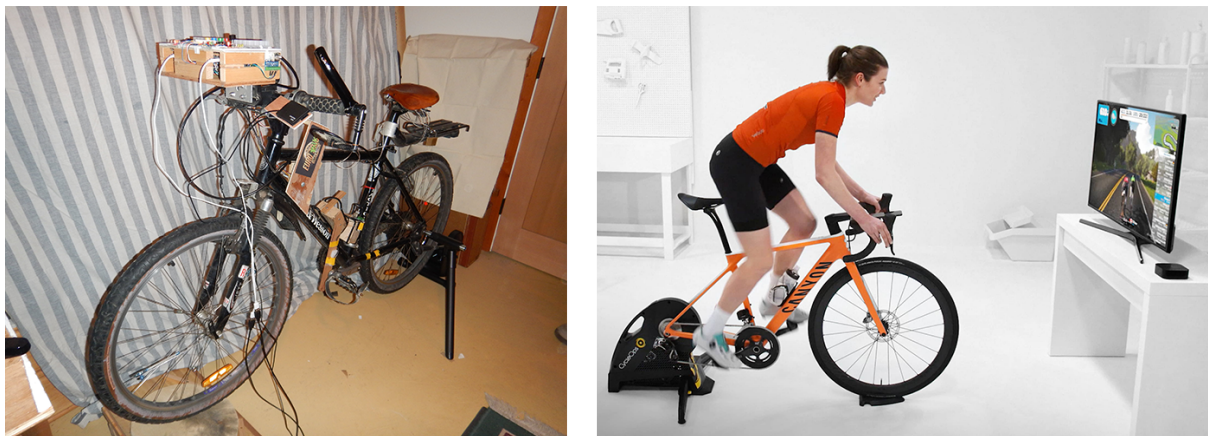
²<https://create.arduino.cc/projecthub/Tazling/usbcycle-ride-through-your-virtual-world-8ff961>

³<https://eu.wahoofitness.com/devices/bike-sensors>

poveternostným podmienkam z pohodlia domova. V aplikácií však chýba možnosť zahájiť závod proti reálnym protivníkom.

2.4 Zwift

Zwift je aplikácia, ktorá poskytuje hru viacerých hráčov, avšak namiesto videí z reálnych oblastí používa virtuálne prostredie pretekov, v ktorom je možné súťažiť s reálnymi používateľmi. Hráč má k dispozícii aktuálne zobrazenie stavu pretekov. Tie sú vizualizované virtuálnymi cyklistami, ktorých pozície sa menia podľa reálnych dát získaných z hardvérových zariadení cyklotrenažérov. Rôzne spoločnosti sa snažia riešiť aj problém zakupovania tohto hardvéru, preto vytvárajú centrá, ktoré sú vybavené potrebnými zariadeniami a ľudia v nich môžu pretekať na prenájatých cyklotrenažéroch. Na obrázku 1 sú zobrazené projekty USBCycle a Zwift. Obrázky boli prevzaté z oficiálnych stránok Zwift⁴ a z oficiálnych stránok projektu USBCycle⁵



Obr. 1: USBCycle a Zwift

2.5 Zhrnutie

Z prieskumu obdobných riešení sa zistilo, že je možné zvýšiť záujem o bicyklovanie na cyklotrenažéroch vhodnou vizuálizáciou získaných dát. Tieto dáta môžu byť získané priamo z výstupu cyklotrenažéru alebo využitím vlastného senzoru a vývojovej dosky Arduino. Pomocou upevnenia tohto senzoru na koleso cyklotrenažéru by malo byť možné získať dostatočné dáta na spracovanie rýchlosti a výpočet prejdenej vzdialenosti. Pre vizuálizáciu dát by mal byť použitý mobilný telefón kvôli jeho kompaktnosti a veľkému rozšíreniu medzi ľuďmi. Virtuálny závod by mal zobrazovať reálnu pozíciu hráčov spolu s informáciami o svojej jazde.

⁴<https://zwift.com/>

⁵<https://www.hackster.io/Tazling/usbcycle-ride-through-your-virtual-world-8ff961>

3 Cyklotrenažéry

K dispozícii je veľké množstvo rôznych prevedení cyklotrenažérov. Kvalita spracovania a dostupné možnosti cyklotrenažéra priamo ovplyvňujú cenu výsledného produktu. Profesionálne produkty slúžiace vrcholovým športovcom poskytujú väčšie množstvo možností a senzorov kvôli úplnému monitorovaniu a analýze tréningu. Pre bežného používateľa obvykle postačuje jednoduchý cyklotrenažér s cyklopočítačom v rozumnej cenovej relácii. Vo všeobecnosti môžeme cyklotrenažéry rozdeliť do niekoľkých kategórií.

3.1 Rozdelenie podľa ceny

Lacné domáce cyklotrenažéry s minimálnou podporou grafického zobrazovania informácií a jednoduchým senzorom pre meranie rýchlosti otáčok kolesa sú najčastejšou voľbou pre rekreačné domáce cvičenie a pre ľudí, ktorí nie sú ochotní investovať veľké peniaze do cyklotrenažéra. Takéto cyklotrenažéry majú vo väčšine prípadov zabudované mechanické nastavovanie záťaže a poskytujú minimálne informácie o jazde. Táto bakalárska práca je primárne určená práve pre takýto typ cyklotrenažérov. Informácie z týchto zariadení môžu byť získané priamym pripojením na senzor, ktorý sníma otáčky kolesa alebo tento senzor môže byť nahradený vlastným senzorom. Profesionálne cyklotrenažéry určené pre náročných používateľov, fitness centrá alebo profesionálnych cyklistov častokrát podporujú grafické zobrazenie informácií získaných z jazdy, meranie srdcového tepu, elektronické regulovanie záťaže a presnejšie prepočty informácií získaných z cyklotrenažéra. Z takýchto zariadení je po dôkladnej analýze možné získať presnejšie dáta, avšak toto vyžaduje pokročilé znalosti z oblasti elektrotechniky a prístup k technickej dokumentácii daného zariadenia.

3.2 Rozdelenie podľa spôsobu prevedenia

Cyklotrenažéry ako samostatná jednotka, ktorá je vybavená šlapacím mechanizmom, regulátorom záťaže a jednoduchým počítačom pre zobrazovanie informácií sú najčastejšou voľbou amatérskych cyklistov a nenáročných používateľov. Bicykle, vybavené tzv. trénerom, ktorý primárne poskytuje stabilitu použitému bicyklu a umožňuje použitie reálneho bicykla na vnútorné cvičenie častokrát slúžia aj na čítanie a odosielanie informácií či na regulovanie záťaže. Tieto zariadenia sa pohybujú vo vyšších cenových kategóriách a bývajú častejšou voľbou profesionálnych športovcov. Špeciálnym typom trénera sú cyklistické válce. Umožňujú vnútorné použitie klasického bicykla pomocou váľčiek, na ktoré sú umiestnené kolesá bicykla. Pri pedálovaní sa váľčky točia protichodne kolesám bicykla čo spôsobuje, že cyklista pedáluje na mieste. Tento typ trénera však neposkytuje zdieľanie informácií o jazde a je iba lacnejšou alternatívou klasického cyklotrenažéra.

Klasické cyklotrenažéry ale aj bicykle vybavené trénerom poskytujú rôzne spôsoby vytvárania záťaže pri pedálovaní. Vytváranie záťaže je dôležité pre zvyšovanie intenzity cvičenia a

k dosahovaniu lepších výsledkov. Jednotky, ktoré sa používajú na generovanie záťaže sa delia do kategórií podľa toho, akým spôsobom záťaž poskytujú. Najčastejšími spôsobmi sú:

- Vzdušný - Jednotka používa ventilátor, ktorý je poháňaný silou pedálovania a poskytuje záťaž zadnému kolesu. Záťaž sa zvyšuje s cyklistovou rýchlosťou a navodzuje pocit bicyklovania na reálnom povrchu. Je však veľmi hlučný a má obmedzenú záťaž.
- Magnetický - Magnetický zotrvačník vytvára záťaž na zadnom kolese. Je veľmi tichý avšak má hornú hranicu poskytovania záťaže.
- Kvapalinový - Kombinuje magnetický zotrvačník s kvapalinovými odporovými komorami. Je veľmi tichý avšak opakovaným zahrievaním a rozpínaním kvapaliny môže dôjsť k poškodeniu tesnenia a jej úniku.
- Odstredivý - Špeciálne navrhnuté odstredivé tlakové dosky poskytujú záťaž pomocou odstredivej sily. Je to veľmi tichý spôsob a používateľ má možnosť prispôbovať si záťaž.

3.2.1 Použité cyklotrenažéry

Pre testovanie funkčnosti mobilnej aplikácie a meranie rýchlosti boli použité dva cyklotrenažéry. Prvým z testovaných cyklotrenažérov je Bremshey Arrow Ambition, ktorý môžeme zaradiť do kategórie lacných domácich cyklotrenažérov, ktoré poskytujú iba základné informácie o jazde a mechanické nastavovanie záťaže. Ku kolesu brzdy je pripevnený senzor s dvojvodičovým vedením, ktorý poskytuje informácie o frekvencii otáčania kolesa. Tie sú ďalej spracované v jednoduchom cyklopočítači a zobrazené používateľovi. Druhý cyklotrenažér môžeme zaradiť do kategórie profesionálnych cyklotrenažérov pre fitness centrá. Ide o cyklotrenažér od spoločnosti Life Fitness LifeCycle 9500 HR, ktorý je vybavený magnetickou záťažou, ktorá je riadená elektronicky pomocou používateľských vstupov na riadiacej doske. Taktiež je vybavený alternátorom, ktorý poskytuje elektrickú energiu pre napájanie elektroniky cyklotrenažéru. Nachádza sa na ňom aj merač tepovej frekvencie a používateľovi poskytuje rozšírené informácie o jeho tréningu. Pre identifikáciu výstupov signálu, ktoré vedú od brzdového mechanizmu cyklotrenažéra je potrebná technická dokumentácia. V oboch prípadoch je navyše použité zariadenie, ktoré zabezpečuje spracovanie informácií a ich prenos do mobilnej aplikácie.

4 Prevod signálu z cyklotrenažéru

Keďže sa na cyklotrenažéri nenachádza žiadne rozhranie, na ktoré by bolo možné pripojiť počítač a čítať užitočné informácie, signál musí byť získaný z výstupov na doskách plošných spojov cyklotrenažéru, alebo musí byť na cyklotrenažér umiestnené vlastné zariadenie pre čítanie informácií o RPM. Pri pripojení na existujúce výstupy signálu je častokrát potrebné vykonať dodatočnú úpravu signálu a správny prepočet signálu na požadovanú veličinu. Keďže technická dokumentácia takéto informácie neobsahuje, neostáva nič iné, ako sledovať reálne informácie z cyklopočítača a porovnávať ich s výstupným signálom. Táto metóda môže viesť k skresleniu výsledku chybou, ktorá bola spôsobená ľudským faktorom. Pri zapojení vlastného zariadenia pre čítanie informácií o RPM môže taktiež dôjsť k skresleniu výsledku kvôli absencii informácie o zvolenej záťaži. Túto informáciu je takmer nemožné získať najmä z lacných cyklotrenažérov, na ktorých sa nastavenie záťaže vykonáva manuálne. Pri elektronicky ovládaných cyklotrenažéroch táto informácia ovplyvňuje práve prepočet aktuálnej rýchlosti vzhľadom na rýchlosť RPM.

4.1 Arduino UNO

Či už sa informácie o jazde získavajú priamo z výstupov cyklotrenažéru, alebo je nutné použiť vlastné zariadenie, je potrebné tieto informácie čítať a spracovávať. Vývojová doska Arduino bola pre projekt vybraná ako ideálna voľba kvôli svojim malým rozmerom, nízkej cene a možnosti spracovania ako analógového, tak aj digitálneho signálu. Toto je dôležité pre použitie Bluetooth modulu HC-05 a pre sledovanie napäťovej úrovne na výstupe cyklotrenažéru. Arduino

- je malý mikrokontrolér, na ktorý môžu byť za pomoci externých súčastiek a niekoľkých pripojovacích pinov zapojené napríklad motory, relé, svetelné senzory, laserové diody a iné,
- môže byť napájané pomocou USB portu, 9V batérie alebo externého zdroja,
- môže komunikovať s počítačom alebo je možné ho po nahraní kódu odpojiť a spustiť samostatne. [7]

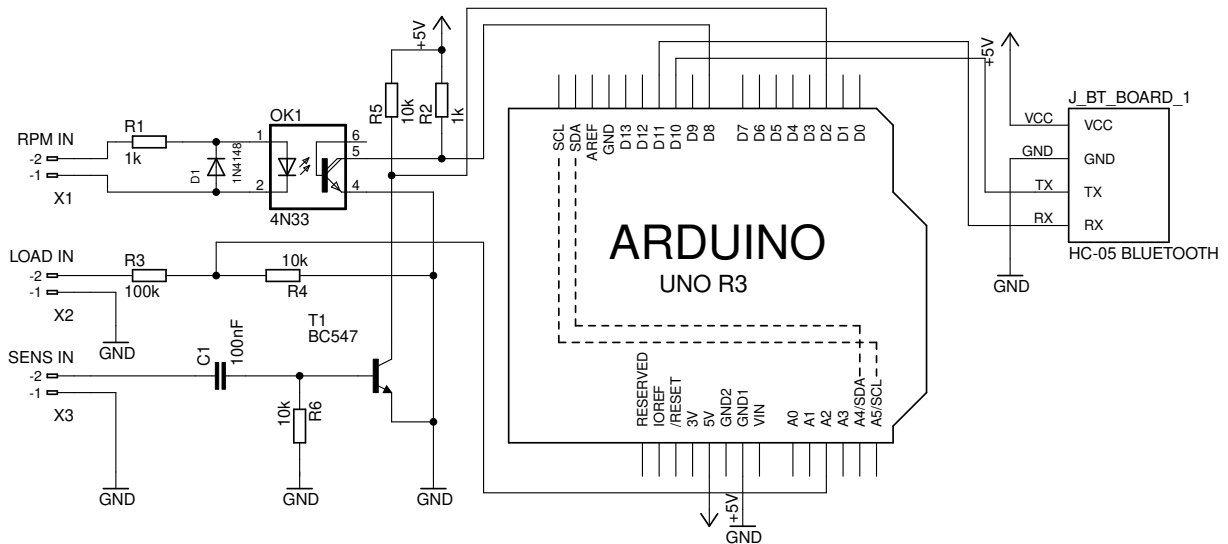
Pre programovanie Arduina sa používa vývojové prostredie Arduino IDE a kód, ktorý je napísaný pre vývojovú platformu Arduino sa nazýva Sketch. Samotný program musí obsahovať iba dve funkcie - funkciu setup, ktorá slúži na definovanie vstupov a výstupov a funkciu loop, ktorá sa spúšťa pravidelne počas toho, ako je Arduino pripojené k zdroju elektrickej energie. Tieto funkcie sú návratového typu void a teda nevracajú žiadny výsledok. Na rozdiel od iných funkcií ich nevolá programátor, ale systém Arduina. Úlohou programátora je iba zadefinovať tieto funkcie v programe aby systém vedel, aké operácie má vykonávať. Pre pripojenie Arduina k počítaču sa používa USB kábel. Na vývojovej doske arduino UNO sa nachádza 14 digitálnych I/O pinov označených D0-D13 slúžiacich na čítanie a zápis digitálnych signálov a analógové piny označené A0-A5 určené na čítanie analógového signálu, ktorý prevádza na hodnotu medzi 0 a 1023. Arduino dokáže napájať externé zariadenia 3,3 alebo 5 voltami.

4.2 Bluetooth modul HC-05

HC-05 je jednoduchý SPP Bluetooth modul komunikujúci pomocou sériovej komunikácie. Obsahuje dva módy, master a slave, ktoré slúžia na zahájenie a potvrdenie spojenia. Je možné medzi nimi prepínať, nedajú sa však použiť súčasne. Pre nastavenie Bluetooth je možné použiť AT Command mód a pomocou AT príkazov zmeniť jeho konfiguráciu. AT príkazy je možné zadávať napríklad cez príkazový riadok sériového monitora Arduina. Vstupné napätie Bluetooth modulu HC-05 je 5V a preto je ideálny pre použitie s Arduino UNO. Na obrázku 2 môžeme vidieť pripojenie modulu HC-05 k Arduino UNO. Modul HC-05 máme nakonfigurovaný a prepnutý do módu slave, ktorý poskytuje používateľovi možnosť pripojenia.

Pre prenos informácií do mobilnej aplikácie bol rovnako zvažovaný aj WiFi modul ESP8266, avšak kvôli podpore závodu viacerých hráčov by bolo neefektívne znemožniť používateľovi pripojenie na internet prostredníctvom technológie WiFi. Zapojenie modulu ESP8266 do režimu poskytujúceho pripojenie na lokálnu WiFi a odosielanie dát na zvolený server by vyžadovalo veľké množstvo konfigurácie, ktoré by v konečnom dôsledku nespĺňalo požadovaný efekt. Mohli by vzniknúť problémy s pripojením modulu na WiFi, ktoré by bolo možné riešiť iba manuálnou konfiguráciou modulu. Pri prípadnej implementácii na verejnom mieste by mohol vzniknúť problém autentifikácie hráčov.

4.3 Schéma zapojenia prevodníku



Obr. 2: Schéma zapojenia prevodníku

Napájanie celého riešenia je realizované cez napájací konektor dosky Arduino, pričom doporučené napájacie napätie je v rozsahu 7-12V. Vyššie napätie sa nedoporučuje z dôvodu väčších tepelných strát spôsobených väčším úbytkom napätia na lineárnom regulátore napätia použitom na doske Arduino. Bluetooth modul je napájaný z 5V vetvy Arduino. Vzhľadom k snahe

vytvoriť najuniverzálnejšie možné riešenie, navrhnuté zapojenie umožňuje pripojenie niekoľkých typov senzorov. Magnetoindukčný senzor je vhodné pripojiť k Arduinu cez optočlen. Vyhne sa tak problémom s napäťovými špičkami a rôznymi úrovňami signálu. Tento typ senzoru je možno pripojiť na svorku RPM IN. V prípade senzorov, ktoré prevádzajú meranú veličinu na hodnotu napätia postačí pripojenie realizovať bežným odporovým deličom na svorke LOAD IN. Tu je nutné upozorniť na galvanické prepojenie zemí cyklotrenažéru a dosky Arduina, čo ale u cyklotrenažérov, ktoré sú napájané zo SELV zdrojov alebo batérií nie je problém. Posledným typom senzorov, ktoré sa vyskytli na testovaných cyklotrenažéroch boli hallove sondy. Pripojenie navrhnutého riešenia je realizované paralelne s dodávaným počítačom cyklotrenažéru. Signál zo svorky SENS IN je zosilnený na úroveň kompatibilnú so vstupmi Arduina pomocou jednoduchého tranzistorového zosilňovača.

4.4 Cyklotrenažér LifeCycle 9500 HR

Pre získanie potrebných dát bolo potrebné zistiť, ako sú v cyklotrenažéri pripojené jednotlivé komponenty. Zo servisného manuálu cyklotrenažéra⁶ vyplývalo, že informácia o RPM je lineárne mapovaná na napäťovú úroveň pinu RPM. Po pripojení osciloskopu sa však zistilo, že je na výstupe daného pinu generovaný oscilujúci napäťový priebeh.

Aby Arduino dokázalo spracovať tento signál, bol navrhnutý jednoduchý obvod pomocou optočlenu, ktorý signál upravil do spracovateľnej formy. Schému zapojenia obvodu môžeme vidieť na obrázku 2.

Do digitálneho vstupu Arduina D8 sme privádzali signál z RPM výstupu cyklotrenažéru a použili sme jednoduchý Sketch na sledovanie striedy meraného signálu a výpočet frekvencie. Vo výpise 1 je možné vidieť implementáciu funkcie pulseIn, ktorá slúži na meranie striedy digitálneho signálu. Funkcia číta pulz v stave logická 0 alebo 1 na zvolenom pine.

```
unsigned long pulseIn(uint8_t pin, uint8_t state, unsigned long timeout)
{
    uint8_t bit = digitalPinToBitMask(pin);
    uint8_t port = digitalPinToPort(pin);
    uint8_t stateMask = (state ? bit : 0);

    unsigned long width = 0;
    unsigned long numloops = 0;
    unsigned long maxloops = microsecondsToClockCycles(timeout) / 16;

    while ((*portInputRegister(port) & bit) == stateMask)
        if (numloops++ == maxloops) return 0;

    while ((*portInputRegister(port) & bit) != stateMask)
```

```

    if (numloops++ == maxloops) return 0;

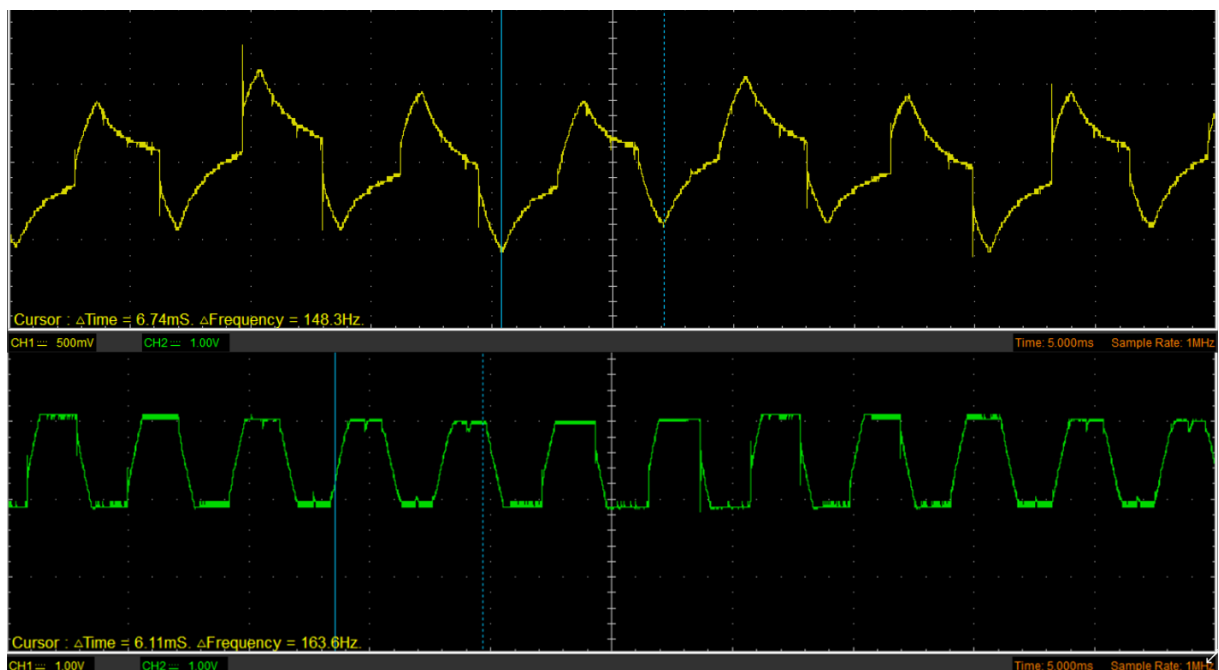
    while ((*portInputRegister(port) & bit) == stateMask) {
        if (numloops++ == maxloops) return 0;
        width++;
    }
    return clockCyclesToMicroseconds(width * 21 + 16);
}

```

Výpis 1: Funkcia pulseIn pre meranie frekvencie na digitálnom pine

Ak je hodnota logická 0, funkcia počká, kým sa pin dostane na hodnotu logickej 1 a začne počítať čas, ktorý uplynú, kým sa pin dostane späť na hodnotu logickej 0. Výsledkom je dĺžka pulzu v mikrosekundách alebo 0, ak funkcia nedokázala prečítať žiadny kompletný pulz. Funkcia funguje pre pulzy od 10 mikrosekúnd až do 3 minút. Funkcia prijíma 2 povinné a 1 voliteľný parameter. Parameter pin je číslo pinu, na ktorom chceme merať striedu, state je stav, ktorý chceme merať a timeout je čas v mikrosekundách, počas ktorého musí byť pulz dokončený. Funkcia clockCyclesToMicroseconds na základe inštrukčných cyklov procesora určuje čas v mikrosekundách. Funkcia microsecondsToClockCycles zabezpečuje opačný prevod.

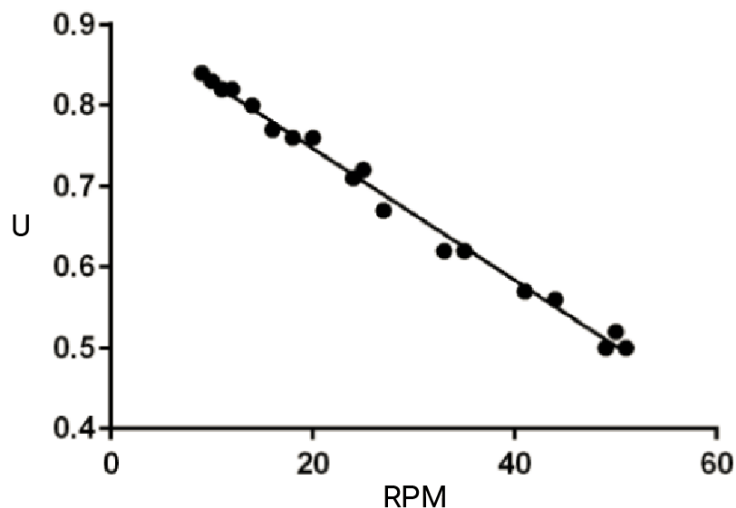
Na obrázku 3 môžeme vidieť porovnanie priebehov surového signálu, ktorý vychádza priamo z PINu RPM popisovaného cyklotrenažera a signálu upraveného pomocou obvodu s optočlenom. Signály na týchto obrázkoch pochádzajú z dvoch rozdielnych meraní.



Obr. 3: Pôvodný signál vychádzajúci z PINu RPM v porovnaní so signálom modifikovaným pomocou optočlenu

Zistilo sa že frekvencia je lineárne mapovaná na RPM, preto bolo pomocou Arduina možné urobiť jednoduchý prepočet nameranej prekvencie na RPM. Prepočet bol vykonaný sledovaním a zapisovaním hodnôt frekvencie v porovnaní s aktuálnou rýchlosťou RPM zobrazenou na displeji cyklotrenažéra. Počas prepočtu však bolo zistené, že informácia RPM vôbec nie je viazaná na reálnu rýchlosť, ktorú cyklotrenažér zobrazoval, ale určovala už spomínanú kadenciu pedálovania, z ktorej nie je možné odvodiť reálnu rýchlosť. Keďže vo virtuálnych pretekoch používateľa nezaujímajú kadencia pedálovania ale rýchlosť a teda prejdená vzdialenosť, bolo potrebné zistiť, na akom pine sa nachádza informácia, ktorá je viazaná na rýchlosť pedálovania a zároveň úroveň záťaže. Cyklotrenažér má elektronicky riadenú záťaž a tento faktor ovplyvňuje výpočet výslednej rýchlosti. Zo servisného manuálu ⁶ bolo zistené, že na pine LOADCMD sa nachádza informácia viazaná na aktuálne zvolenú záťaž. Meraním sa zistilo, že na pine sa nachádza jednosmerné napätie, ktoré je mapované na RPM podľa aktuálne zvolenej záťaže. Na výstupnom pine sa však nachádzalo napätie až 12V a vzhľadom na TTL logiku Arduina a modulu HC-05 bolo potrebné vykonať zníženie napätia pomocou jednoduchého napäťového deliča. Výstupný signál sa delí v pomere 1:10, čo zabezpečí bezpečnú napäťovú úroveň pre Arduino. Schéma tohto napäťového deliča je zobrazená na obrázku 2.

Sledovaním rýchlosti vzhľadom na zvolenú záťaž a rýchlosť RPM sa podarilo pomocou lineárnej regresie odvodiť vzorec, ktorý sa dá použiť na prevod vstupného signálu. Lineárna regresia je matematická metóda ktorá predstavuje aproximáciu daných hodnôt priamkou metódou najmenších štvorcov. Jej výsledkom je tzv. regresná priamka, ktorá je zobrazená na obrázku 4.



Obr. 4: Priamka lineárnej regresie

Čierne body zobrazené na obrázku 4 boli namerané behom experimentu a predstavujú napäťové úrovne na výstupe cyklotrenažéra viazané na rýchlosť RPM. Na osi U sa nachádzajú namerané úrovne napätia z výstupu cyklotrenažéra znížené pomocou deliča napätia z obrázku 2

⁶<https://www.manualslib.com/manual/692348/Life-Fitness-Lifecycle-9500hr-Series.html>

a na osi RPM je zobrazená rýchlosť RPM. Keďže poznáme napätie, dokážeme pomocou rovnice vypočítať výslednú rýchlosť. Výsledná rovnica pre prepočet modifikovaného analógového signálu na rýchlosť je

$$Y = -126,5 * U + 114,8 \quad (1)$$

kde Y je výsledná rýchlosť a U je namerané napätie.

Výsledná hodnota je počítaná v hlavnej slučke programu. Aby bolo zamedzené náhlým výkyvom napätia alebo nepredvídateľným anomáliám, vzorky sú zbierané v skupinách, v rámci ktorých sa počíta aritmetický priemer nameraných hodnôt, ktorý je odosielaný do mobilnej aplikácie.

5 Vývoj mobilných aplikácií

Hlavnou časťou tejto bakalárskej práce je vytvorenie funkčnej aplikácie, ktorá používateľom poskytuje pripojenie na cyklotrenažéry vybavené potrebným hardvérom popisovaným v kapitole 4. Voľba platformy pre vývoj aplikácie bola ovplyvnená veľkou popularitou mobilných telefónov, ich kompaktnosťou a spoľahlivou použiteľnosťou. Mobilné telefóny sa dnes veľmi približujú funkciám počítačov. Sú však pohodlnejšie na používanie a sú vhodným adeptom nie len pre jednoduché aplikácie. Rôzne komerčné riešenia využívajú počítačové aplikácie, avšak kvôli jednoduchosti riešenia a nízkym požiadavkam na výkon je vhodnejšia voľba mobilnej platformy. Voľbu mobilnej platformy Android podporil aj fakt, že v druhom kvartáli roku 2016 bol podľa ⁷ výskyt zariadení s OS Android (86,2%) v porovnaní s iOS (12,9%) a ostatnými (0,9%).

5.1 Použité technológie

Aplikácia je vyvíjaná v jazyku Java, ktorý je jedným z doporučených jazykov pre programovanie platformy Android. Java poskytuje hlavné knižnice pre programovanie platformy Android a veľkú základňu programátorov, ktorí píšú vlastné knižnice a zjednodušujú tak ďalší vývoj. Pre programovanie je použité vývojové prostredie Android Studio, ktoré poskytuje stabilné prostredie pre vývoj a je oficiálnym IDE pre vývoj Androidu. Zároveň obsahuje množstvo preddefinovaných funkcionalít, ktoré zjednodušujú proces programovania a skracujú čas potrebný pre vývoj. Aplikácia je optimalizovaná pre mobilné telefóny a tablety a podporuje pripojenie zariadenia na cyklotrenažér pomocou technológie Bluetooth. Prihlásenie používateľa je zabezpečené pomocou služby Google Play Games, ktorá poskytuje programové zabezpečenie hier pre Android. Pre využívanie hry pre viacerých hráčov musí byť používateľ v službe Google Play Games zaregistrovaný a prihlásený. Táto služba pre správne fungovanie vyžaduje pripojenie k internetu, preto je potrebné, aby mal používateľ zapnutú technológiu WiFi alebo mobilné dáta a bol pripojený k internetu. V opačnom prípade bude používateľovi za predpokladu, že je pripojený na cyklotrenažér, umožnené sledovať len aktuálne informácie o jeho jazde, nie však vytvárať virtuálne závody a pozývať priateľov. Aplikácia má pôsobiť jednoducho a prehľadne s rozložením prvkov typických pre moderné aplikácie platformy Android. Minimálna verzia Android podporujúca túto aplikáciu je API level 21. V texte sa ďalej budú nachádzať anglické výrazy, ktoré nejde dobre preložiť do Slovenčiny a preto sú ponechané pôvodné názvy, ktoré budú pri prvom použití skrátené vysvetlené, ak to bude vyžadovať správne pochopenie kontextu.

5.2 Cross-platform aplikácie

Pre jednoduchšiu prenositeľnosť aplikácie sa dostávame k možnosti vývoja cross-platformových aplikácií. Jednou z dostupných možností pre riešenie problémov prenositeľnosti je tzv. cross-compilation. Cross-compiler oddeľuje vývojové prostredie od cieľového prostredia. Vo vývoji

⁷<https://android.jlelse.eu/apple-vs-android-a-comparative-study-2017-c5799a0a1683>

mobilných aplikácií to funguje nasledovne: framework poskytuje platformovo nezávislé API používajúc programovacie jazyky ako napr. JavaScript, Ruby alebo Java. Vývojári používajú API na vytváranie mobilnej aplikácie vrátane používateľského rozhrania, dátového úložiska a business logiky. Kód je spracovaný cross-compilerom, ktorý ho transformuje do platformovo špecifických natívnych aplikácií [3].

Ďalšou možnosťou vývoja cross-platformových mobilných aplikácií je využitie mobilných webových technológií. Toto zahŕňa používanie štandardných webových technológií ako HTML, CSS a JavaScript pre vytvorenie aplikácie, ktorá sa správa ako natívna aplikácia. Je to možné vďaka možnostiam HTML5 a CSS3 zahŕňajúc vstavané SQL databázy, lokálne úložisko, animácie, canvas, web sockety a prehrávanie videa. Webové aplikácie bežia ako samostatný webový prehliadač alebo v prehliadačovom zobrazení vstavanom v natívnej aplikácii (Hybridný web). V tomto hybridnom modeli je webová aplikácia spustená v tenkom wrapperi, ktorý poskytuje premostenie s operačným systémom zariadenia a jeho službami. Komunikácia medzi webovou aplikáciou a natívnou aplikáciou je zabezpečená pomocou JavaScriptu. Táto technika sa pokúša priniesť to najlepšie z oboch technológií: flexibilitu webových aplikácií s rýchlosťou a možnosťami natívnej aplikácie. Najznámejšími frameworkmi pre vývoj cross-platformových aplikácií sú Rhodes, Corona alebo Phonegap.

Ďalšími zaujímavými alternatívami sú frameworky ako Xamarin alebo Unity, ktoré taktiež podporujú vytváranie cross-platformových aplikácií. Pri Unity nie je potrebné vytvárať samostatné projekty pre viaceré cieľové platformy. Nie je preto problém začať s vývojom aplikácie cielennej na stolové počítače a neskôr v priebehu vývoja prepnúť cieľovú platformu na Android alebo iOS bez nutnosti zmeny kódu alebo rozloženia aplikácie [2].

Po dôkladnom zvážení a prediskutovaní bol vývoj aplikácie pre túto bakalársku prácu pomocou cross-platformových technológií zamietnutý a pre vývoj bol zvolený jazyk Java.

5.3 Java pre Android

Java nie je iba objektovo orientovaný jazyk ale je to taktiež súbor technológií, ktoré robia vývoj softvéru rýchlejší a výsledné aplikácie sú robustnejšie a bezpečnejšie. Java je vhodná práve pre svoju platformovú nezávislosť, jednoduchosť použitia, bezpečnosť a škálovateľnosť [5]. Programy Javy sú kompilované pomocou Java kompilera ktorý sa označuje ako Javac. Kým javac dokáže skompilovať zdrojové kódy Javy do bytekódu, pre spustenie takého programu na Androide je potrebné použiť ART. Ten sa prvýkrát objavil vo verzii KitKat a stal sa štandardom vo verzii Lollipop. [9]. Software, ktorý zjednodušuje vývoj aplikácie sa nazýva IDE. Na trhu je veľké množstvo IDE ako napríklad Eclipse, NetBeans alebo Android Studio. Android Studio, ktoré je oficiálnym nástrojom pre vývoj platformy Android, bolo použité aj na vývoj aplikácie pre túto bakalársku prácu.

Pri voľbe platformy v prípade tejto bakalárskej práce hrala najväčšiu rolu veľká používateľská základňa vývoja aplikácií v Jave pre Android, jej rýchlosť a priama podpora Google Play Games Services, ktorá výrazne zjednodušila implementáciu hry pre viacerých hráčov. Pri mul-

tiplatformových aplikáciách by sa znížila rýchlosť behu aplikácie a tým by sa znížil aj celkový dojem a hrateľnosť, zatiaľ čo tieto faktory sú pri vývoji tejto aplikácie kritické.

5.4 Zaradenie aplikácie

Väčšina aplikácií pre Android spadá do jednej z nasledujúcich kategórií.

- Foreground je to aplikácia, ktorá je užitočná len ak je v popredí a je pozastavená ak nie je viditeľná. Najlepším príkladom tohto typu aplikácie sú hry.
- Background je aplikácia, ktorá má obmedzenú interakciu s používateľom a okrem doby, počas ktorej je nastavovaná, strávi väčšinu svojho času skrytá na pozadí. Tieto aplikácie sú menej obvyklé, avšak dobrými príkladmi môže byť budík alebo automatické odpovedanie na SMS správy.
- Intermittent - väčšina správne navrhnutých aplikácií spadá do tejto kategórie. Jedným extrémom sú aplikácie, ktoré očakávajú limitovaný čas na interaktivitu s používateľom a zvyšok času bežia na pozadí. Ďalším extrémom sú aplikácie, ktoré zvyčajne bežia na popredí, ale dôležitú časť práce vykonávajú na pozadí.
- Widgety a Live pozadia - niektoré aplikácie sú reprezentované iba ako widgety alebo ako Live pozadia, ktoré poskytujú interaktívne vizuálne komponenty priamo na domovskej obrazovke [6].

Komplexné aplikácie väčšinou nespádajú do jedinej kategórie, ale využívajú elementy z viacerých typov. Naša aplikácia sa radí z najväčšej časti do Foreground aplikácií, keďže po zastavení nevykonáva žiadnu činnosť a celý beh aplikácie je spojený s interakciou s používateľom. Na pozadí funguje iba služba, ktorá zabezpečuje komunikáciu s Bluetooth zariadením a Broadcast Reciever, ktoré čakajú na vopred definovaný typ správy. Služba, ktorá zabezpečuje komunikáciu s Bluetooth zariadením je ukončená, iba ak ju používateľ zastaví manuálne, vznikne chyba pri prenose Bluetooth alebo ak je ukončená celá aplikácia. Pri stave, kedy je aplikácia pozastavená, služba funguje na pozadí a je vypnutá iba pri zavolaní metódy `onDestroy`. Ak by došlo k nutnosti minimalizovať aplikáciu alebo by bolo vyvolané náhodné prerušenie ako je napr. prichádzajúci hovor a používateľ bol pripojený k fitness zariadeniu, nie je potrebné sa znovu pripájať a komunikácia je zachovaná.

6 Bluetooth

Bluetooth je spôsob, akým dokážu zariadenia bezdrôtovo komunikovať na krátke vzdialenosti. Je definovaný štandardom IEEE 802.15.1. a spadá do kategórie osobných počítačových sietí. Kladie špeciálny dôraz na komunikáciu na menej ako cca. 9 metrov. Komplexný súbor dokumentov nazývaný Bluetooth Specifications popisuje do detailov všetko od základného rádiového signálu až po najvyššiu úroveň protokolov pre bezdrôtovú komunikáciu s krátkym dosahom. Od toho, aký typ spojenia bude nadviazaný závisí, ktoré zariadenie odošle prvý dátový paket aby bolo zahájené spojenie a komunikácia. Zariadenia zahajujúce odchádzajúce spojenie musia pred vytvorením spojenia a prenášaním dát zvoliť transportný protokol. Zariadenia vytvárajúce prichádzajúce spojenie musia zvoliť transportný protokol a pred potvrdením spojenia a prenášaním dát musia začať počúvať. Každý Bluetooth čip, ktorý bol kedy vyrobený, je označený globálne unikátnou 48 bitovou adresou označovanou ako Bluetooth adresa alebo adresa zariadenia identická MAC adresa v technológii Ethernet. Keďže by bolo pre ľudí obtiažne vysporiadať sa so 48 bitovým číslom, Bluetooth zariadenia takmer vždy obsahujú aj používateľsky prívetivé meno zariadenia. Toto meno je bežne zobrazované pre identifikáciu zariadenia, avšak pre pripojenie k zariadeniu sa používa adresa zariadenia. Pre väčšinu zariadení je meno zariadenia nastaviteľné v používateľskom rozhraní [4]. Objavovanie zariadení je proces pre vyhľadávanie blízkych Bluetooth zariadení. Pre zistenie dostupných zariadení sa posiela Broadcast discovery správa. Broadcast správa sa väčšinou používa pre kontaktovanie všetkých dostupných zariadení. Detailnejšie informácie sa zo zariadení získavajú po individuálnom kontaktovaní každého zo zariadení. Čas ktorý prejde počas vykonávania pripojenia zariadení je najväčšou nevýhodou Bluetooth. Vyhľadávanie okolitých zariadení môže trvať približne 5 až 10 sekúnd a samotné pripojenie môže v extrémnych prípadoch dosiahnuť čas až 15 sekúnd.

Bluetooth Low Energy sa stalo súčasťou Bluetooth 4.0 špecifikácie. Pre pripojenie BLE zariadení definujeme dve roly: Central(master) a Peripheral(slave). Zariadenie v role Master skenuje pripojiteľné pakety v prednastavených frekvenciách a rozhoduje o zahájaní spojenia. Slave periodicky odosiela pakety a potvrdzuje prichádzajúce pripojenia. Akonáhle je pripojenie aktívne, slave zariadenie sleduje centrálné časovanie a pravidelne si vymieňa dáta [10]. Technológia BLE v našej aplikácii nie je využitá, naskytuje sa však ako možnosť pri ďalšom vývoji aplikácie.

6.1 Využitie pri prenose signálu

Pre pripojenie k zariadeniu, ktoré sníma dáta z cyklotrenažéra používame Bluetooth modul HC-05. Toto zariadenie bolo vybrané pre svoju jednoduchosť, nízku cenu a pomerne jednoduchú konfiguráciu. Posledné špecifikácie Bluetooth však definujú BLE, ktoré by sa pre použitie na účel pravidelného odosielania dát z cyklotrenažéra do aplikácie javilo taktiež ako zaujímavá alternatíva, ktorej by sa pri prípadnom rozširovaní tohto projektu mala venovať značná dávka pozornosti.

6.2 Bluetooth v Android aplikácií

Technológia Bluetooth je v návrhu aplikačnej logiky implementovaná ako služba. Služby v OS Android sú podobné službám alebo démonom v stolových alebo serverových operačných systémoch [8]. Sú to spustiteľné časti kódu, ktoré zvyčajne bežia na pozadí od času, kedy sú inštanciované až kým sa aplikácia nevypne. Zvyčajne neposkytujú používateľské rozhranie a môže byť nutné použiť ich, ak je v aplikácii potreba vykonávať úlohy na pozadí bez používateľského rozhrania. Iné aplikačné komponenty môžu spustiť službu a tá pokračuje v behu na pozadí aj keď sa používateľ prepne na inú aplikáciu⁸. Dodatočne sa komponenta môže naviazať na službu a spolupracovať s ňou, dokonca vytvárať medziprocesovú komunikáciu IPC.

V návrhu aplikačnej logiky sa vyhľadávanie a pripájanie na zariadenie vykonáva v hlavnej aktivite, ktorá slúži aj ako hlavné menu pre používateľa. V ňom má možnosť zahájiť vyhľadávanie zariadení a pripojiť sa na požadované zariadenie. Po úspešnom pripojení sa hráčovi sprístupnia rôzne typy závodov, podľa stavu pripojenia na Google Play Games Services. Táto funkcionálna je implementovaná v hlavnej aktivite kvôli tomu, že je vykonávaná iba ako jednorázová akcia a vykonáva sa výlučne z tejto aktivity. Samotná komunikácia medzi Bluetooth modulom HC-05 už beží na pozadí ako samostatná služba, ktorá je spúšťaná bezprostredne po pripojení na zvolené zariadenie. Pripájanie na nepodporované zariadenia je ošetrené už pri samotnom vyhľadávaní. V samostatnej triede je udržiavaný zoznam adries zariadení, ktoré boli použité v zariadeniach pre spracovanie signálu z cyklotrenažérov. Nepodporované zariadenia sú automaticky vyfiltrované ešte pred zobrazením dostupných zariadení používateľovi. Po úspešnom pripojení na podporované zariadenie sa zaháji príjem informácií z cyklotrenažéra v preddefinovanom formáte, ktorý je v aplikácii rozložený na užitočné informácie. V Bluetooth službe je vytvorená samostatná trieda v ktorej je spustené vlákno čakajúce na príchod správy z Bluetooth modulu. Po príchode tejto správy je pomocou funkcie `sendBroadcast` vyvolaná akcia, ktorej reakciou je rozloženie na užitočné informácie. Broadcast receiver, ktorý má za úlohu počúvať na akékoľvek zmeny stavu Bluetooth a vyvolávať k nim príslušné reakcie je zadaný v samostatnej triede a spustený z rodičovskej aktivity, z ktorej dedia všetky ostatné aktivity aplikácie. Pri strate synchronizácie s Bluetooth zariadením je používateľ upozornený s možnosťou znovupripojenia alebo ukončenia závodu. Kedykoľvek sa používateľ nachádza v hlavnom menu, má možnosť odpojiť sa od zariadenia a pripojiť sa k inému, ak toto zariadenie nie je pripojené na iné zariadenie. V jednu chvíľu môže byť na zariadenie, ktoré odosiela informácie z cyklotrenažéra pripojené iba jedno zariadenie. Po úspešnom pripojení alebo odpojení je používateľ informovaný aplikačnou správou a sú mu sprístupnené alebo zablokované rôzne typy závodov. Vo výpise 2 môžeme vidieť časť kódu, ktorý má za úlohu získať z nájdeného zariadenia adresu a nadviazať spojenie s týmto zariadením. Najskôr sa pokúsime získať Bluetooth Adapter a ak je Bluetooth dostupné, z nájdeného zariadenia prečítame jeho adresu a pomocou nej sa na zariadenie pripojíme. V opačnom prípade je služba zastavená.

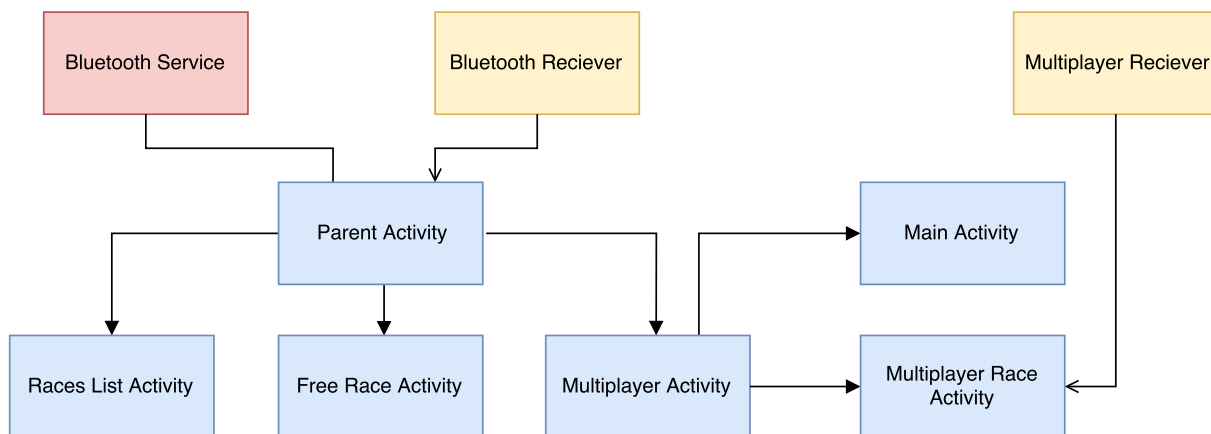
⁸<https://developer.android.com/guide/components/services.html>

```
mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
    if (mBluetoothAdapter != null) {
        device = mBluetoothAdapter.getRemoteDevice(address);
        deviceName = device.getName();
        String macAddress = device.getAddress();
        if (macAddress != null && macAddress.length() > 0) {
            connectToDevice(macAddress);
        } else {
            stopSelf();
            return Service.START_STICKY_COMPATIBILITY;
        }
    }
    String stopservice = intent.getStringExtra("stopservice");
    if (stopservice != null && stopservice.length() > 0) {
        stop();
    }
    return START_STICKY;
```

Výpis 2: Nadviazanie spojenia s Bluetooth zariadením

7 Návrh aplikácie

Návrh aplikácie pozostáva z funkčnej a grafickej časti. V grafickej časti sú v zložkách definovaných vo vývojom prostredí Android Studio zotriedené grafické prvky. Rozloženie prvkov grafickej časti aplikácie je umiestnené v zložke layout. V nej sú definované pohľady na všetky aktivity a iné časti aplikácie. Kvôli úspore uložených dát a taktiež z hľadiska používateľského zážitku je v aplikácii použitá vektorová grafika. Všetky použité obrázky boli prevedené pomocou nástrojov Android Studia do XML súborov. Vo formáte rastrovej grafiky je v aplikácii použitá iba ikona aplikácie a ikona hráča, ktorá sa získava zo služby Google Play Games. Z hľadiska funkčnej stránky je aplikácia rozdelená do zložiek, podľa kategórií služieb Androidu, ktoré budeme využívať. Na pozadí sú v prípade potreby spúšťané služby, ktoré slúžia na vykonávanie operácií a zber informácií bez nutnosti použitia používateľského rozhrania. Taktiež sú v aplikácii použité ListView, ktorých jednotlivé adaptéry sú definované v samostatnej zložke. Pre zobrazovanie informácií v dialógových oknách sú použité DialogFragmenty, ktoré slúžia ako medzikroky medzi jednotlivými aktivitami pre nastavenie vlastností aktivít alebo informovanie používateľa. Služby alebo aktivity častokrát musia kontaktovať iné aktivity a pre tento účel sú v aplikácii použité Broadcast Recievery, ktoré sledujú prichádzajúce správy a vykonávajú určitú časť kódu podľa typu prichádzajúcej správy. Kvôli logickému rozdeleniu aplikácie a sprehľadneniu kódu boli vytvorené aj pomocné statické triedy pre funkcie, ktoré sa v aplikácii často používajú a bolo by nezmyselné vytvárať ich viackrát, alebo vytvárať inštanciu triedy. Pomocou statickej triedy je vytvorený aj zoznam pripojiteľných zariadení, ktorý je taktiež potrebné získavať z rôznych častí kódu. Pre ukladanie štruktúrovaných dát boli vytvorené triedy, ktoré zabezpečia ich konzistentné uloženie, sprehľadnia prácu v kóde a zjednodušia získavanie informácií z objektov.



Obr. 5: Zjednodušený diagram komponentov aplikácie

Na zjednodušenom diagrame 5 môžeme vidieť základné komponenty rozdelenia aplikačnej logiky. Červenou farbou je vyznačená služba, ktorá komunikuje s rodičovskou aktivitou. Žltou farbou sú označené Broadcast Recievery a modrou farbou sú samotné aktivity.

7.1 Aktivita

Aktivita sa dáajú zrovnávať s pomocnými programami na stolových systémoch, ako sú napríklad kancelárske aplikácie. Sú to časti spustiteľného kódu, ktoré sú inštanciované používateľom alebo operačným systémom a sú spustené pokiaľ sú vyžadované. Dokážu vykonávať interakciu s používateľom a získavať dáta alebo služby z iných aktivít alebo služieb pomocou dotazov alebo Intentov [8]. Väčšina spustiteľného kódu napísaného pre Android bude spustená v kontexte aktivity. Aktivita zvyčajne odpovedajú obrazovke displeja a pokiaľ práve nie sú aktívne, môžu byť zničené operačným systémom pre uvoľnenie pamäte. Ďalej môžeme povedať, že Android aplikácia môže pozostávať z jednej alebo z niekoľkých aktivít [1]. Aktivita sa správa ako kontajner pre používateľské rozhranie a kód zabezpečuje jeho fungovanie. Na aktivity sa môžeme pozeráť ako na stránky aplikácie - jedna stránka v aplikácii predstavuje jednu aktivitu.

Aktivita v našej aplikácii sú rozdelené do niekoľkých úrovní. Najvyššou aktivitou je *ParentActivity*, ktorá dedí z *AppCompatActivity* a obsahuje metódy a atribúty, ktoré sú potrebné takmer vo všetkých ostatných aktivitách a bolo by nezmyselné, vytvárať ich vždy nanovo. Obsahuje metódy a atribúty pre prístup k lokálnemu úložisku, zobrazovaniu a skrývaniu dialógov počas načítania rôznych funkcionálít aplikácie, synchronizáciu Bluetooth, filtrovanie rýchlostí získavaných z cyklotrenažéru a iné. Filtrovanie rýchlostí je potrebné vykonávať hlavne kvôli možným výkyvom a extrémnym hodnotám získaných z cyklotrenažéru. Funkcia sa v pravidelných intervaloch snaží získať informácie o rýchlosti a ak zistí nelogické výkyvy v rýchlostiach, ktoré môžu byť spôsobené náhlym zastavením alebo snahou začať závod v inej, ako nulovej rýchlosti, používateľ je okamžite vyzvaný, aby svoju rýchlosť stabilizoval pod určenou hodnotou. Až po úspešnej synchronizácii rýchlosti je používateľovi zahájené zobrazovanie nameranej rýchlosti a počítanie prejdenej vzdialenosti. Z *ParentActivity* dedia takmer všetky ostatné aktivity v aplikácii a ďalej definujú špecifickejšie chovanie. Aktivita, ktoré nededia z *ParentActivity* sú aktivity, ktoré zobrazujú úvodné oboznámenie s aplikáciou a animáciu, ktorá je zobrazená pri každom spustení aplikácie. Tieto aktivity dedia z aktivít definovaných v knižniciach tretích strán. Aktivita, ktoré priamo dedia z *ParentActivity* sú *Multiplayer* a *RacesListActivity*. *Multiplayer* slúži ako rodičovská aktivita pre aktivity, ktoré majú podporovať komunikáciu s Google Play Games Services. Sú nimi aktivity *MultiplayerRace* a *MainActivity*. *MainActivity* je spúšťaná bezprostredne po zapnutí aplikácie a zobrazení úvodnej animácie. Táto aktivita slúži zároveň ako hlavné menu, kde sa spúšťajú všetky dostupné závody, vykonávajú sa rôzne akcie a slúži ako východiskový bod pre všetky ostatné časti aplikácie. Zabezpečuje nám taktiež vyhľadávanie a nadviazanie spojenia s Bluetooth zariadením, definuje nám chovanie všetkých elementov obsiahnutých v rozložení aktivity a slúži aj na vyhodnocovanie ukončených aktivít a získavanie prenesených informácií. Zabezpečuje nám aj prihlásenie používateľa pomocou Google Play Games a umožňuje zahájenie akcií, ktoré súvisia s hrou pre viacerých hráčov ako sú napríklad vyhľadávanie priateľov, reakcia na pozvánku alebo zahájenie vlastného závodu pre viacerých hráčov. *MultiplayerRace* je aktivita, ktorá sa stará o funkčnosť akéhokoľvek závodu pre viacerých hráčov. Keďže dedí z aktivity

Multiplayer, obsahuje všetky potrebné metódy pre nadviazanie spojenia a komunikáciu medzi hráčmi. Taktiež sa stará o synchronizáciu dát medzi používateľmi a prípadnú voľbu závodu.

RacesListActivity je priamym potomkom *ParentActivity* a slúži na zobrazovanie lokálne uložených závodov vo forme *ListView*. Pre správne naplnenie je zadefinovaný vlastný adaptér, ktorý má za úlohu správne rozložiť a zobrazit dáta v *ListView*. Pri jednotlivých závodoch je možné zobrazit podrobnosti o závode alebo daný závod odstrániť. Poslednou aktivitou, ktorá priamo dedí z *ParentActivity* je *FreeRace*. Je to samostatný typ tréningu, pri ktorom je potrebné byť pripojený na Bluetooth fitness zariadenie. Používateľ má v tomto móde možnosť sledovať svoje fitness štatistiky. Nie je potreba implementovať Google Play Games Services a tento typ tréningu je možné zahájiť aj v tom prípade, ak používateľ nie je prihlásený do Google Play Games.

7.2 Intenty a Broadcast Recievery

Intenty sú používané ako mechanizmus pre prenos správ ktorý funguje ako v rámci aplikácie, tak aj medzi viacerými aplikáciami [6]. Intenty môžeme použiť nasledujúco:

- Explicitne spustiť Službu alebo aktivitu použitím názvu triedy.
- Spustiť aktivitu alebo službu a vykonať nejakú akciu s konkrétnym typom údajov.
- Broadcastovať vykonanie nejakej udalosti.

Jedným z najčastejších použití Intentov je vytvorenie novej aktivity explicitne (pomocou názvu triedy) alebo implicitne (definovaním akcie na konkrétny typ dát). Intenty sa dajú použiť aj na broadcastovanie správ do systému. Aplikácie dokážu zaregistrovať Broadcast Recievery ktoré čakajú a reagujú na Broadcast Intenty. Toto nám umožňuje vytvoriť aplikáciu založenú na aplikačných a systémových udalostiach, alebo na udalostiach tretích strán. Pre posielanie Intentov pomocou metódy `sendBroadcast` by sa mala Intentu nastaviť akcia, dáta a kategória, aby Broadcast Recievery dokázali rozpoznať ich význam. V tomto prípade sa pre identifikáciu broadcastu používa unikátny action string, ktorý by mal byť vytvorený podobne, ako sa vytvárajú názvy Java balíčkov.

Broadcast Recievery musia byť zaregistrované v kóde alebo v aplikačnom manifeste. Pri použití Recieverov, ktoré sú definované v manifeste, môže byť Reciever vykonaný aj keď aplikácia práve nie je spustená.

V našej aplikácii definujeme viaceré typy Intentov a aplikačne zaregistrované Broadcast Recievery, ktoré reagujú na udalosti vyvolané počas behu služby, ktorá sa stará o pripojenie na Bluetooth zariadenia a na udalosti, ktoré sú vyvolané počas hry pre viacerých hráčov. Vo výpise 3 môžeme vidieť registráciu nového Broadcast recieveru, ktorý zabezpečuje prijímanie dát o hre pre viacerých hráčov.

```
multiplayerReciever = new MultiplayerReciever(this);
    IntentFilter multiplayerFilter = new IntentFilter();
    multiplayerFilter.addAction(getString(R.string.intent_mult_msg));
    registerReceiver(multiplayerReciever, multiplayerFilter);
```

Výpis 3: Multiplayer Broadcast reciever

Broadcast Reciever, ktorý reaguje na udalosti vyvolané počas používania služby pre Bluetooth pripojenie definuje podľa prenesených typov správ nasledujúce stavy:

- Správa prijatá - ak Reciever dostal informáciu o prijatej správe, pokúsi sa túto správu rozložiť a nastaviť aktuálnu rýchlosť na novú hodnotu.
- Zmena stavu - pri získaní tejto informácie sa rozlišujú ešte dodatočné stavy. Pri zmene stavu na stav pripájanie je zobrazené dialógové okno s načítavaním. Ak bol stav zmenený na stav pripojený, vykonajú sa nastavenia používateľského rozhrania a zruší sa dialógové okno načítania. Pri zmene stavu na odpojený sú zmenené dáta v používateľskom rozhraní a znemožnené pripojenie na závody.
- Strata pripojenia - pri strate pripojenia je používateľovi zobrazené dialógové okno s možnosťou opätovného pripojenia alebo zrušenia závodu. V prípade opätovnej straty sa táto možnosť opakuje.

Broadcast Reciever reagujúci na zmeny stavov hry pre viacerých hráčov definuje stavy ako prihlásenie a odhlásenie používateľa do závodu, vytvorenie nového závodu spojené s nastavením konkrétneho typu závodu, iniciovanie závodu, aktualizácia skóre ostatných protivníkov a ukončenie závodu z dôvodu odpojenia potrebného počtu hráčov pre pokračovanie v hre. Všetky akcie sú spojené s aktivitou *FreeRace* a aplikačná logika ako aj vytvorenie Broadcast Recieveru je viazané na túto aktivitu.

7.3 Pomocné funkcie

Funkcie, ktoré sú v aplikácii potrebné na rozdielnych miestach by bolo neefektívne deklarovať na každom mieste zvlášť. Ide o funkcie, ktoré zabezpečujú prevod ID jednotlivých závodov na reálne názvy, a na názvy súborov drawable, ktoré sa majú v konkrétnom type závodu použiť. Ďalšie funkcie slúžia na serializáciu objektov na String a deserializáciu Stringov na objekty. Pri inštanciách tried, ktoré chceme serializovať musíme nastaviť interface triedy na Serializable. Pri serializácii objektov musíme vytvoriť z objektu bytové pole a pomocou kódovania BASE64 vytvoriť výsledný String. Celá funkcia je obalená do try-catch bloku, ktorý zabezpečuje vytvorenie výnimky v prípade, že počas serializácie vznikne chyba.

```
public static String serializeToString(Object object){
    try {
        ByteArrayOutputStream bo = new ByteArrayOutputStream();
        ObjectOutputStream so = new ObjectOutputStream(bo);
        so.writeObject(object);
        so.flush();
        return new String(Base64.encode(bo.toByteArray(), Base64.NO_WRAP));
    } catch (Exception e) {
        e.printStackTrace();
    }
    return null;
}
```

Výpis 4: Serializácia objektu na String

Pri deserializácii Stringu na objekt využívame opačný postup a pomocou vytvoreného bytového poľa definujeme výsledný objekt, ktorý musí byť následne pretypovaný na špecifický typ objektu.

Ďalšie pomocné funkcie slúžia na prevody časových a zobrazovacích jednotiek, jednotiek vzdialenosti a synchronizáciu rýchlosti. Všetky tieto funkcie sú definované ako statické funkcie a pre ich použitie nie je potrebné mať vytvorenú inštanciu triedy, v ktorej sú funkcie definované.

7.4 Používateľské rozhranie

View je hlavným elementom používateľského rozhrania Androidu. Je to obdĺžniková oblasť na obrazovke, ktorá má na starosti vykresľovanie a reagovanie na udalosti. View je základnou stavebnou jednotkou používateľského rozhrania Androidu podobne ako odstavec textu <p> alebo odkaz <a> sú základnými stavebnými jednotkami HTML stránky. Niektorými zo základných View v Androide sú TextView, ImageView, Layout a Button [1].

V našej aplikácii je pre dôveryhodnejší zážitok z hry nastavený mód na celú obrazovku, pre ktorý je využitý tzv. immersive mód a skrytý ActionBar, ktorý môžeme vidieť v klasických Android aplikáciách. Pri každom spustení aplikácie je najskôr spustená úvodná animácia s logom a názvom aplikácie.

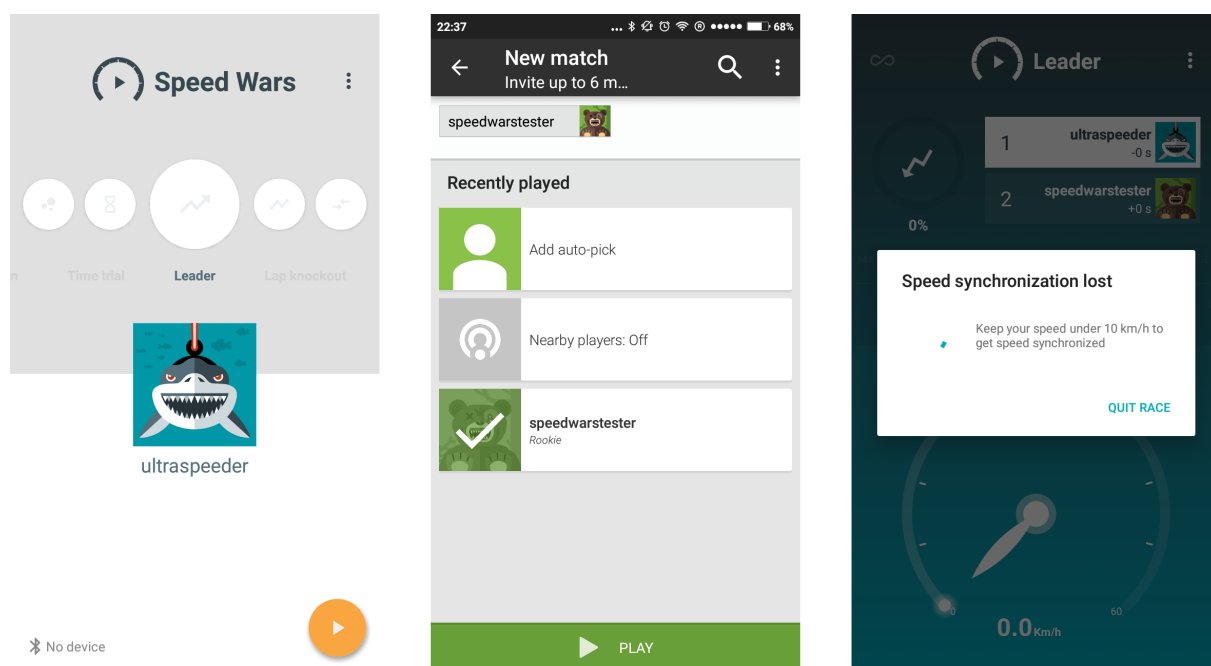
Pre prvotné oboznámenie používateľa s aplikáciou je pri jej prvom spustení, alebo neskôr na vyžiadanie používateľa spustená informačná časť aplikácie, ktorá má za úlohu ukázať používateľovi, ako aplikácia funguje a poskytnúť základné informácie o zahájení závodov.

Používateľské rozhranie pozostáva z hlavného menu, ktoré slúži na základné nastavenia závodov, pripájanie na Bluetooth zariadenie a prihlásenie alebo odhlásenie z Google Play Games. Na hlavnej obrazovke sa nachádza aj dialógové menu, v ktorom môžeme spustiť históriu závodov, ktorá je lokálne uložená na zariadení alebo môžeme vykonať ukončenie aplikácie. Po úspešnom

pripojení mobilného telefónu na Bluetooth fitness zariadenie je používateľovi umožnené zahájenie voľného tréningu a pri úspešnom pripojení do služby Google Play Games mu je umožnené aj vytvorenie všetkých ostatných typov závodov, prípadne vytvorenie rýchleho závodu. Pri prichádzajúcej pozvánke od iného používateľa je pozvánka používateľovi zobrazená formou dialógového okna v obrazovke hlavného menu. Ak nastane situácia, kedy je aplikácia pri prichádzajúcej pozvánke na začatie závodu ukončená, pozvánka príde formou notifikácie zo služby Google Play Games a po kliknutí na pozvánku je spustená aplikácia a zahájené pripájanie na Bluetooth zariadenie a následné spustenie závodu.

Ďalšou časťou používateľského rozhrania sú aktivity vygenerované pomocou Google Play Games, ktoré slúžia na nastavovanie hry pre viacerých hráčov. Tieto aktivity budú ďalej popísané v sekcii 8.1.

Samostatnú časť tvorí voľný tréning, v ktorom sú zobrazené štatistiky o cvičení, aktuálna rýchlosť a menu, v ktorom je možné ukončiť závod a zastaviť alebo sputiť automatické striedanie zobrazovaných informácií. Podobné informácie sú aj v závode pre viacerých hráčov, navyše však má zobrazovaciu časť, v ktorej je zobrazený aktuálny stav závodu, čas alebo vzdialenosť do ukončenia závodu a pozícia jednotlivých hráčov. Na obrázku 6 môžeme vidieť ukážku hlavného menu, herného prostredia a voľbu protivníkov.



Obr. 6: Hlavné menu voľba protivníkov a závod

8 Google Play Games Services

Spoločnosť Google poskytuje vývojárom herné služby pomocou ktorých je možné vytvárať hry, ktoré obsahujú ocenenia, rebríčky, multiplayer, udalosti a možnosť uloženia rozohraných hier. V našej aplikácii je využitá služba Real-time Multiplayer, ktorá zabezpečuje komunikáciu medzi účastníkmi závodu.

8.1 Realtime Multiplayer

Vývojár môže pomocou API Real-time Multiplayer prepojiť viacerých hráčov v rámci jedného herného sedenia a prenášať dáta medzi pripojenými používateľmi⁹. Používanie tejto API môže pomôcť zjednodušiť herný vývoj a rieši niekoľko dôležitých úloh.

- Spravuje sieťové pripojenie a udržiava tzv. Real-time multiplayer Room čo je virtuálna konštrukcia, ktorá umožňuje sieťovú komunikáciu medzi viacerými hráčmi v jednom hernom sedení a umožňuje posilať dáta priamo medzi hráčmi.
- Poskytuje hráčom výber ostatných hráčov na pripojenie do herného sedenia, automatické spárovanie alebo ich kombináciu.
- Ukladá uchádzačov a stav sedenia na serveroch Google Play Games Services počas životného cyklu Real-time multiplayer hry.
- Posiela pozvánky a aktualizácie hráčom a notifikácie zobrazuje na všetkých zariadeniach na ktorých je používateľ prihlásený.

Implementáciu Real-time multiplayer služby môžeme rozdeliť do niekoľkých krokov. Najskôr je potrebné inicializovať Room, s možnosťou konfigurácie úloh, ktoré v hre daný používateľ zastáva. Keďže v našich závodoch sú všetci používatelia rovnocenní, nemusíme ďalej definovať špeciálne parametre. Pre spracovanie informácie o pozvánkach máme v aplikácii definovaný InvitationCallback, ktorý nám zabezpečuje spracovanie pozvánok a akcie s nimi spojené. Pre spracovanie udalostí súvisiacich s našou Room definujeme RoomStatusUpdateCallback, ktorý má na starosti pripájanie alebo odpájanie účastníkov a niekoľko ďalších pomocných udalostí, ktoré v tomto scenári môžu nastať. Aby sme vedeli spracovať udalosti, ktoré sú vyvolávané pri vytváraní, pripájaní a odpájaní od Room, musíme definovať aj RoomUpdateCallback. Všetky tieto Callbacky spolu s ďalšími pomocnými funkciami nám umožňujú plnohodnotné využívanie Real-time multiplayeru. Všetky informácie sú Broadcastované a prijímané cez už spomínaný MultiplayerReceiver. Pri spúšťaní náhodného výberu závodu a automatického párovania hráčov je nutné programovo vyriešiť, kto bude iniciátor závodu, hoci navonok sa účastníci tvária ako rovnocenní. Na toto nám slúži pomocná funkcia iAmTheKing, ktorá rozhodne o tom, či bude daný účastník iniciátorom závodu alebo nie, keďže iniciátor je vždy len jeden.

⁹<https://developers.google.com/games/services/common/concepts/realtimeMultiplayer>

Funkcia vezme zoznam účastníkov závodu a pred jeho začatím zotriedi všetkých účastníkov podľa jedinečného kľúča, ktorým je meno ako identifikátor v Google Play Games. Ak sa po zotriedení nachádzame na prvom mieste zoznamu, sme iniciátorom závodu a v používateľskom rozhraní sa nám odomkne tlačidlo pre zahájenie závodu.

```
public boolean iAmTheKing(){
if (actualMultiplayerRaceType == MULTIPLAYER_RACE_TYPE_QUICK_MATCH){
    Collections.sort(mParticipants, new Comparator<Participant>() {
        @Override
        public int compare(Participant p1, Participant p2) {
            return p1.getDisplayName().compareToIgnoreCase(p2.
                getDisplayName());
        }
    });
    if (mParticipants != null && mParticipants.get(0).getDisplayName().
        equals(mPlayer.getDisplayName())){
        return true;
    }
    return false;
}
else if (actualMultiplayerRaceType ==
    MULTIPLAYER_RACE_TYPE_MATCH_CREATOR){
    return true;
}
return false;
}
```

Výpis 5: Voľba iniciátora závodu

Ak počas závodu vznikne situácia, kedy nie je možné v zápode pokračovať z dôvodu nedostatku hráčov, tento závod je ukončený v prospech posledného aktívneho hráča. Komunikácia a odosielanie správ o stave závodu je vykonávané pomocou zasielania správ v rámci služby Google Play Games Services. Pre bežné odosielanie správ v hlavnej hernej slučke sa používajú tzv. unreliable messages, ktoré negarantujú doručenie informácie do cieľa. Informácia o finálnom skóre je zasielaná pomocou reliable messages ktoré doručenie do cieľa garantujú. Pomocou týchto metód dokážeme poslať pole bytov v ktorom je uložená užitočná informácia.

Google Play Games Services nám podstatne zjednodušilo implementáciu podpory hry pre viacerých hráčov. Predpripravené Intenty zabezpečujú menšiu šancu vnesenia chyby a spoľahlivo fungujú pre výber protivníkov, vytvorenie Room alebo správu pozvánok. Výrazne nám pomohla aj vzorová implementácia Real-time multiplayer hry, ktorú Google poskytuje voľne na stiahnutie.

9 Testovanie

Hotová aplikácia bola testovaná na už spomínaných cyklotrenažéroch Bremshey Arrow Ambition a LifeCycle 9500 HR. Pozorovaním sme zistili, že výsledné hodnoty zobrazované v aplikácii odpovedajú reálnym hodnotám zobrazovaným na zabudovanom cyklopočítači s presnosťou približne 5%. Problém však je, že hodnota zobrazovaná na cyklopočítači je na pohľad nepresná. Použitý cyklotrenažér LifeCycle 9500 HR prepočítava aktuálnu rýchlosť vzhľadom k elektricky nastavenej záťaži a pri pedáľovaní v nižších záťažových stupňoch tento cyklotrenažér poskytuje nepresné informácie. Rýchlosť sa pri týchto nízkych záťažových stupňoch takmer vôbec nemení ani pri zvyšovaní kadencie pedáľovania.

Aplikácia bola zavedená do produkčného kanála Google Play a je voľne dostupná na stiahnutie na adrese ¹⁰ pod názvom Speed Wars. Cyklotrenažér LifeCycle 9500 HR spolu s ďalším zapožičaným cyklotrenažérom z fitness centra pre účely tejto bakalárskej práce bol vybavený zariadením pre prenos rýchlosti do mobilnej aplikácie a tak bolo využívanie aplikácie prístupné aj širokej verejnosti.

¹⁰<https://play.google.com/store/apps/details?id=com.gymify.speedwars>

10 Záver

Cieľom práce bolo zvýšiť motiváciu a záujem o bicyklovanie na cyklotrenažéroch vytvorením a vhodnou vizualizáciou virtuálneho závodu medzi viacerými používateľmi formou aplikácie pre OS Android, s čím súvisí aj spracovanie a prenos informácií z cyklotrenažéra do aplikácie. Podarilo sa nám analyzovať a interpretovať signál prichádzajúci z cyklotrenažeru LifeCycle 9500 HR. Analýza informácií prebehla formou odčítania údajov počas tréningu a následnou interpoláciou informácií pomocou lineárnej regresie. Podarilo sa nám vytvoriť aj mobilnú aplikáciu podporujúcu hru viacerých hráčov v reálnom čase. Prenos signálu do mobilnej aplikácie prebehol pomocou Arduino Uno a Bluetooth modulu HC-05. Taktiež sa nám podarilo vytvoriť snímač otáčok cyklotrenažera Bremshey Arrow Ambition a pomocou Arduino Nano a Bluetooth modulu HC-07 prenášať rýchlosť do mobilnej aplikácie. V aplikácii sú lokálne uložené všetky ukončené závody.

Vývoj nových technológií a funkcie Google Play Games Services poskytujú ďalšie možnosti pre vývoj tejto bakalárskej práce. Ako prvé sa naskytuje použitie technológie BLE, ktorá by zvýšila životnosť batérie. Ďalej sa naskytuje aj použitie WiFi technológie pre prenos informácií do aplikácie, čo by ale vyžadovalo množstvo bezpečnostných opatrení a nastavovania modulu, ktorý by zabezpečoval prenos signálov na určené servery. Bolo by taktiež nutné vytvoriť a nastaviť vlastný server, ktorý by bol prispôbený na prenos určených údajov a pripájanie používateľov. Ušetrilo by to však nutnosť pripájania používateľa na zariadenie pomocou Bluetooth a jedinou nutnosťou by ostalo pripojenie na internet pomocou WiFi alebo mobilných dát.

V aplikácii by bolo možné pridať ďalšiu službu Google Play Games Services - Leaderboards. Leaderboards sú rebríčky, v ktorých by sa mohli ukladať najlepšie výkony používateľov v rôznych disciplínach. Taktiež by sa mohli použiť Achievements pre získavanie ocenení za predom určené dosiahnuté limity. Pri nútenom prerušení hry by sa rovnako mohla použiť služba Saved Games, ktorá zabezpečuje uloženie aktuálneho stavu hry a možnosť pokračovať v hre neskôr.

V porovnaní s ostatnými dostupnými riešeniami je naša aplikácia použiteľná na takmer každý cyklotrenažér, avšak pri niektorých cyklotrenažéroch je potreba zložitejšej analýzy. Aplikácia poskytuje jednoduché používateľské rozhranie a pre hru viacerých hráčov vyžaduje iba nastavenie účtu Google Play Games. Aplikácia je oproti komerčným riešeniam dostupná v obchode Google Play zadarmo. Nedisponuje však realistickou vizualizáciou závodu a preto nemusí byť pre náročnejších používateľov dostatočne pútavá. Oproti spomínaným domácim projektom v sekcii 2 poskytuje jednoduchšie zapojenie a prenos signálu.

V rámci testovania aplikácie boli dva cyklotrenažéry vo fitness centre vybavené zariadeniami pre prenos informácií o rýchlosti do mobilnej aplikácie a bolo tak umožnené testovanie aplikácie širokej verejnosti. Tieto zariadenia budú vo fitness centre ďalej ponechané a zároveň bude formou hodnotení na Google Play sledovaná odozva od ľudí, ktorí aplikáciu vyskúšali. Aplikácia je voľne dostupná pod názvom Speed Wars pre zariadenia s OS Android.

Literatúra

- [1] Michael Burton and Donn Felker. *Android App Development For Dummies*. John Wiley & Sons, 2015.
- [2] Thomas Finnegan. *Learning Unity Android Game Development*. Packt Publishing Ltd, 2015.
- [3] Gustavo Hartmann, Geoff Stead, and Asi DeGani. Cross-platform mobile development. *Mobile Learning Environment, Cambridge*, 16(9):158–171, 2011.
- [4] Albert S Huang and Larry Rudolph. *Bluetooth essentials for programmers*. Cambridge University Press, 2007.
- [5] B. Kurniawan. *Java for Android, Second Edition*:. Brainy Software Incorporated, 2015.
- [6] Reto Meier. *Professional Android 4 application development*. John Wiley & Sons, 2012.
- [7] Simon Monk. *Programming Arduino: getting started with sketches*. McGraw Hill Professional, 2016.
- [8] Rick Rogers, John Lombardo, Zigurd Mednieks, and Blake Meike. *Android application development: Programming with the Google SDK*. O'Reilly Media, Inc., 2009.
- [9] Jonathan Simon. *Head first Android development*. O'reilly, 2011.
- [10] Kevin Townsend, Carles Cufí, Robert Davidson, et al. *Getting started with Bluetooth low energy: tools and techniques for low-power networking*. Ö'Reilly Media, Inc.", 2014.

Adresárová štruktúra priloženého disku

\Arduino - Obsahuje Arduino Sketch pre odosielanie informácie z cyklotrenažéra

\Android - Obsahuje kompletný projekt pre OS Android

\Android-apk - Obsahuje inštalovateľný súbor aplikácie pre OS Android

\Ostatné - Obsahuje stiahnuté kópie webových stránok použitých v tejto bakalárskej práci